

d his

(FILE 'USPAT' ENTERED AT 19:19:51 ON 17 FEB 1998)
L1 140 S CALL? AND PERSONAL DIGITAL ASSISTANT
L2 18 S L1 AND DIRECTORY
L3 10 S L2 AND SEARCH
L4 754 S SEARCH (P) DIRECTORY
L5 3 S L4 AND L3

=> d ab 1-

US PAT NO: 5,715,395 [IMAGE AVAILABLE]

L5: 1 of 3

ABSTRACT:
Disclosed is a apparatus and method for reducing resource location traffic in a compute network. The reduction in location traffic is obtained by a method and apparatus such that a node which has initiated a **search** for a resource which cannot be found starts a timing cycle interval during which subsequent initiating requests at the node are automatically failed without performing the network **search**. This reduces network traffic for searches that are likely to fail. Also disclosed is a threshold counter that alleviates possible difficulties that this may cause for high demand resources. The threshold counter is incremented each time a **search** for a specific resource is automatically failed. A network **search** is performed when either the interval expires or the threshold counter exceeds a threshold count.

US PAT NO: 5,689,547 [IMAGE AVAILABLE]

L5: 2 of 3

ABSTRACT:
In a method of storing **directory** information in a cellular radiotelephone, the cellular radiotelephone system is provided with a network **directory** database including a plurality of telephone numbers. The user is prompted for the input of **search** criteria, and the input **search** criteria is accepted. A **search** request data signal is generated in response to the input **search** criteria, and this signal is sent to the cellular system. The network **directory** database is searched for a match with the **search** criteria, and one or more telephone numbers can be identified. These telephone numbers are returned to the radiotelephone and stored in a memory of the radiotelephone. This method eliminates the need to interact with a **directory** assistance operator and reduces the time of connection between the radiotelephone and the cellular system.

US PAT NO: 5,604,492 [IMAGE AVAILABLE]

L5: 3 of 3

ABSTRACT:
A method and apparatus for displaying **directory**-linked canned messages on a display (20) of a portable device (10) having selective call receiving functions. A personal name **directory** (100) is stored in the device (10). The name **directory** (100) comprises a plurality of entries (102), each entry comprising at least one telephone number and at least one canned message corresponding to the at least one telephone number. A paging message including a telephone number is received by the portable device (10). The personal name **directory** (100) is searched to determine if the telephone number received in the paging message matches with a telephone number in any of the personal

1
name **directory** entry (102). If there is a match, a canned message
which corresponds to the personal name **directory** entry that matches
the telephone number received in the paging message is displayed.
Otherwise, a default message is displayed.

d kwic

US PAT NO: 5,715,395 [IMAGE AVAILABLE]

L13: 1 of 1

ABSTRACT:

Disclosed . . . The reduction in location traffic is obtained by a method and apparatus such that a node which has initiated a **search** for a resource which cannot be found starts a timing cycle interval during which subsequent initiating requests at the node are automatically failed without performing the network **search**. This reduces network traffic for searches that are likely to fail. Also disclosed is a threshold counter that alleviates possible difficulties that this may cause for high demand resources. The threshold counter is **incremented** each time a **search** for a specific resource is automatically failed. A network **search** is performed when either the interval expires or the threshold counter exceeds a threshold count.

SUMMARY:

BSUM(8)

Nodes . . . programs, tasks, processes, operating systems, or hardware. The network addressable unit may be a terminal, workstation, personal computer, mobile computer, **personal digital assistant** or any other hardware. For instance a printer may be a network user that receives instruction(s), file(s), message(s) from a . . .

SUMMARY:

BSUM(24)

Accordingly, . . . The reduction in location traffic is obtained by a method and apparatus such that a node which has initiated a **search** for a resource which cannot be found starts a timing cycle interval during which subsequent initiating requests at the node are automatically failed without performing the **search**. This reduces network traffic for searches that are likely to fail. In the preferred embodiment the present invention also alleviates possible difficulties that this may cause for high demand resources by having a threshold counter which is **incremented** each time a **search** for a specific resource is automatically failed. A network **search** is performed when either the interval expires or the threshold counter exceeds a threshold count.

DETDESC:

DETD(24)

The present invention can reduce network location traffic in variety of ways when used with the network **search** or locate procedure. In the preferred embodiment the present invention performs an Unavailable Resource Table (URT) **check** prior to the network server node performing step 40 as shown in FIG. 2. In an alternative embodiment the **check** for a previous network **search** is performed by the server NN after the server NN has searched its own domain. However, the present invention can be implemented before or after the server NN domain **search**. The **check** may be also be performed before or after any of the steps shown in FIG. 2. In general the present . . . on the server

NNs before any searching is the preferred embodiment of the present invention. The URT, unavailability period and **search** threshold mechanisms are described in detail below.

DETDESC:

DETD(31)

The . . . containing the resource identifier associated with the target resource). The time is entered into the URT entry, after the network **search** for the target resource has failed or been terminated or when a determination has been made that the target resource. . . field of the URT entry associated with the target resource. The unavailability period field of the URT entry is then **checked** when a subsequent request for the same target resource is received by the originating node before a network **search** is commenced. When this **check** is performed, if the current time exceeds the time in the unavailability period field associated with the requested target resource or the unavailability period field is empty, then a network **search** is allowed to commence for the requested target resource. If the current time does not exceed the time in the unavailability period field then a network **search** is not commenced and a failed reply or request or resource unavailable message is returned to the requesting entity. The. . . current time may be the time the request is received by the originating node or the time the URT is **checked** by the originating node or some other time.

DETDESC:

DETD(34)

When, the originating node has initiated a network **search** for a target resource which cannot be found or is unavailable, the originating node may initialize a counter which is updated when subsequent requests for the target resource are automatically failed by the originating node without initiating the network **search**. Like the Unavailability Period, the **search** threshold is a mechanism that is utilized after a network **search** has been unable to determine the location or availability of a requested target resource. Essentially the **search** threshold operates as a counter. It tracks the number of requests made for a target resource after a network **search** for the target resource has failed or terminated. When this count exceeds a THRESHOLD set for the target resource then another network **search** can be commenced for the target resource. The THRESHOLD value is the number of subsequent requests for the resource that will be automatically failed before a network **search** for the target resource is initiated or allowed to commence. If the THRESHOLD is not exceeded then the subsequent request is automatically failed without initiating a network **search** for the target resource and the threshold count is **incremented**. The **search** threshold provides a threshold counter which is **incremented** each time a **search** for a specific resource is automatically failed.

DETDESC:

DETD(35)

The **search** threshold mechanism may be implemented in variety of ways in addition to that described above. One way is to have a **search** threshold field (i.e., threshold counter) associated with each entry in the URT. After a network **search** for the target resource has failed, zero is entered in the **search** threshold field of the URT entry associated with the target resource. This **search** threshold field of the URT entry associated with the target resource is then **checked** when a subsequent request is received for the target resource but, before

another network **search** is commenced for the target resource. If the **search** threshold (i.e., threshold counter) for the target resource exceeds the THRESHOLD (or is equal to the THRESHOLD in another variant) then a network **search** is allowed to commence. If the threshold counter is less than or equal to the THRESHOLD (or less the THRESHOLD in another variant) then a network **search** is not allowed to commence and the locate request is automatically failed and the threshold counter is **incremented**. The THRESHOLD may also be stored in a field in the URT. Another way the threshold counter can be implemented is by initializing the **search** threshold portion of the URT entry with the THRESHOLD for the target resource and decrementing it each time a locate request is automatically failed. In this embodiment the network **search** is only commenced when the threshold counter field is zero.

DETDESC:

DETD(42)

In response to the request 500 but, before initiating a network **search** for the target resource the server NN determines whether a previous network **search** has failed for the target resource. In the preferred embodiment of the present invention this is done by **checking** a URT at the server NN to determine if an entry had been previously made for the target resource.

DETDESC:

DETD(43)

Before **checking** the URT as shown in step 502 the present invention may also include step 501. In step 501 a special flag or indicator contained in the request is **checked** to determine if the URT **check** should be bypassed. If this bypass flag in the request is set to bypass then the URT **check** is not performed and a network **search** or Locate procedure is carried out as shown in step 515. If the flag is not set to bypass then the URT **check** is performed as described below. Bypassing the URT **check** may result in better performance in certain instances such as where a request is received to determine the availability of. . . needs to determine if the target resource is "truly unavailable" (i.e., not unavailable by virtue of any other node's URT **check**). That is the server NN does not want another intermediate node through which the **search** or locate request is being routed to automatically fail the **search** request because of the intermediate nodes own URT **check**. Thus, the bypass flag allows for bypassing URT **check** when it is desirable.

DETDESC:

DETD(47)

In an alternative embodiment to updating the URT entry after **checking** the URT for and finding a target resource that has expired, the entry may be deleted from the URT (i.e., after the entry is deleted a subsequent request for the same target resource, received before the results of the initiated **search** are available, will not find an entry in the URT for the target resource. Thus, the URT **check** is clear. If prevention of overlapping searches for this target resource is desired another **check** is required to prevent this subsequent request from initiating another network **search** for the target resource. Such a **check** to make sure no searches are pending for the target resource might be made by examining the control blocks in. . .

DETDESC:

DETD(60)

After receiving a target resource request the LTR 801 then processes the request. The LTR 801 may **check** a bypass flag in the request. If the bypass flag is set the LTR 801 will bypass the URT **check** and initiate or continue the network **search**. For each request the LTR 801 **checks** whether a previous network **search** was conducted, provided either the bypass flag is not set or the bypass **check** is not performed. This URT **check** is done by the LTR 801 **checking** the URT 805 for an entry that contains a representation of the target resource in the resource identifier field of any entry in the URT. The target resource identifier can be obtained from the request itself. How this **check** function is implemented will depend on the physical structure of the URT. For instance, if the URT is stored in . . . the target resource identifier in the resource identifier field of the entry. If a flat file is used then the **check** may look at each entry until the appropriate entry is found or all entries have been **checked**. Indexing may be used to expedite searching of the URT. Thus, the **check** function may make use of any **search** technique and accompanying data structure.

DETDESC:

DETD(62)

The LTR 801 initiates a network **search** for the target resource based on either the determination that the target resource was not unavailable by the determination function or that no entry was found in the URT 805 for the target resource by the **check** function or that the URT **check** should be bypassed by the bypass **check** function. The Network **Search** function or Locate function 803 carries out the network **search**. As shown in FIG. 8 it may consult one or more directories or caches and forward requests to and receive. . . other network elements 811. A network elements may be an LU, EN, NN, server NN, CD or GN. When the **search** results are obtained from the network **search** function 803 the LTR 801 (or the Network **search** function 803) processes the results/replies to determine if the target resource is unavailable. Based on this determination the LTR 801 updates the URT 805 accordingly and either the LTR 801 or the Network **search** function 803 informs the originating resource of the **search** results as appropriate. The LTR 801 acts as a filter for the Network **search** function 803. The LTR 801 limits the number of network searches initiated by a server NN. By limiting the number. . .

CLAIMS:

CLMS(1)

What . . .

resources and a URT, each URT capable of containing entries, each entry having a resource identifier, unavailability period and a **search** threshold, a method for reducing network resource location traffic comprising the steps of:

- checking** the URT of a first server node for an entry having a resource identifier that contains a representation of a . . . node;
- determining that the target resource is unavailable if the unavailability period of the found entry has not expired and the **search** threshold of the found entry has not been exceeded, if an entry is found in the URT;
- informing the source resource that the target resource is unavailable and updating the **search** threshold of the found entry in the URT, if the target resource is determined to be unavailable;
- initiating a network **search** for the target resource, if no entry was found in the URT or if the target resource was not determined to be unavailable; and

updating the URT based on the results of the network **search**, if the network **search** the target resource is unable to locate the target resource and deletes any entry for the target resource if. . .

CLAIMS:

CLMS (7)

7. . . . having a resource identifier and an unavailability period, a method for reducing network resource location traffic comprising the steps of:

checking the URT of a first server node for an entry having a resource identifier that contains a representation of a. . . the source resource that the target resource is unavailable, if the target resource is determined to be unavailable;

initiating a network **search** for the target resource, if no entry was found in the URT or if the target resource was not determined to be unavailable; and

updating the URT based on the results of the network **search**, if the network **search** for the target resource is unable to locate the target resource and deletes any entry for the target resource if. . .

CLAIMS:

CLMS (11)

11. . . .

unavailable resources, said table having one or more entries each entry containing an resource identifier, an unavailability period and a **search** threshold;

a means for **checking** the URT of the node for an entry having a resource identifier that contains a representation of a target resource,. . . for determining if the target resource is unavailable, said determining means determining that the target resource is unavailable if the **checking** means found an entry and the availability period of the found entry has not expired and the **search** threshold of the found entry has not been exceeded;

means for informing the source resource that the target resource is unavailable if the determining means determines that the target resource is unavailable;

means for initiating a network **search** for the target resource, if no entry was found in the URT or if the target resource was not determined. . . . to be unavailable or if the bypass flag was set; and

means for updating the URT in response to the network **search** results, if the network **search** for the target resource is unable to locate the target resource and deletes any entry for the target resource if. . .

CLAIMS:

CLMS (12)

12. The apparatus of claim 11 wherein prior to the initiating means initiating a network **search** the update means:

updates the entry in the URT, if an entry was found by the **checking** means and either the unavailability period expired or the **search** threshold was exceeded;

creates an entry in the URT for the target resource, if no entry was found by the **checking** means.

CLAIMS:

CLMS (15)

15. . . .

resources, said table having one or more entries each entry containing an resource identifier and an unavailability period and a **search** threshold;

a means for **checking** the URT of the node for an entry having a resource identifier that contains a representation of a target resource, . . . for determining if the target resource is unavailable, said determining means determining that the target resource is unavailable if the **checking** means found an entry and the availability period of the found entry has not expired;

means for informing the source resource. . . the target resource is unavailable if the determining means determines that the target resource is unavailable;

means for initiating a network **search** for the target resource, if no entry was found in the URT or if the target resource was not determined to be unavailable; and

means for updating the URT in response to the network **search** results, if the network **search** for the target resource is unable to locate the target resource and deletes any entry for the target resource if. . .

CLAIMS:

CLMS (16)

16. The apparatus of claim 15 wherein prior to the initiating means initiating a network **search** the update means:

updates the entry in the URT, if an entry was found by the **checking** means and either the unavailability period expired or the **search** threshold was exceeded;

creates an entry in the URT for the target resource, if no entry was found by the **checking** means.

CLAIMS:

CLMS (19)

19. . . .

resources, said table having one or more entries each entry containing an resource identifier and an unavailability period and a **search** threshold;

a means for **checking** the URT of the node for an entry having a resource identifier that contains a representation of a target resource, . . . for determining if the target resource is unavailable, said determining means determining that the target resource is unavailable if the **checking** means found an entry and the availability period of the found entry has not expired;

means for informing the source resource. . . the target resource is unavailable if the determining means determines that the target resource is unavailable;

means for initiating a network **search** for the target resource, if no entry was found in the URT or if the target resource was not determined to be unavailable; and

means for updating the URT in response to the network **search** results, if the network **search** for the target resource is unable to locate the target resource and deletes any entry for the target resource if. . .

d kwic 1, 7, 9, 10, 12, 13

US PAT NO: 5,675,324 [IMAGE AVAILABLE]

L25: 1 of 22

US-CL-CURRENT: 340/825.44; 235/383; 340/311.1; 455/31.2

ABSTRACT:

Next, the location counter indicative of the **called** location of the address directory memory section 14a is cleared (S12). Then, it is determined whether or not the value. . .

DETDESC:

DETD(38)

If not in S13, the processes in S14-S18 is performed with respect to the registered **calling** number at the first location in the location **counter** so as to determine whether or not the registered **calling** number is identical with the received **calling** number. Namely, the **calling** number of the position indicated by the location **counter** is stored in the buffer section 13d (S14), and a identification number determining routine is executed (S15). Then, the facsimile number pointed out by the location **counter** is stored in the buffer section 13d (S16), and the **calling** number identification determining routine is executed (S17). If the **calling** number is determined to be identical, according to the process (to be described later), the registered **calling** number detected by the **search** and the supplemental information attached thereto are displayed on the display section 7. On the other hand, when it is determined that the received **calling** number is not identical with the registered **calling** number, the location **counter** is counted up by 1 (S18), and the sequence goes back to S13 to repeat the described operations.

DETDESC:

DETD(39)

In the described manner, the operations in S13-S18 are repeated. After the **search** of the information is completed to the end of the memory region, if it is determined that no corresponding registered **calling** number is found, it is determined that the received **calling** number is not stored in the address **directory** memory section 14a. This can be confirmed by the fact that the value of the location **counter** is identical with the total number of registrations plus one stored in the total number of registrations memory section 14a. If it is determined in S13 that the value of the location **counter** is identical with the registered **calling** number plus one, the received **calling** number is combined with the fixed message "No corresponding information is found", and the combined information is displayed on the. . .

DETDESC:

DETD(40)

Since the corresponding information is not found in the address directory memory section 14a, in order to store the received **calling** number as new information, the value in the total number of registrations memory section 14b is counted up by +1 (S20), and the received **calling** number is stored in the transmitting and receiving use **calling** number location position, and the received time is stored under the column of the social information (S21), and the alarm sound is beeped so as to inform the person holding the paging device that the **calling** number is received (S22).

DETDESC:

Next, . . . the processes in S24-S30 are repeated in the same manner as S5-S11, and the number of characters in the registered **calling** number and the location of the symbol stored in the buffer section 13d are detected. Here, different from the first. . . the second counter section 13e (see FIG. 9). If it is determined that the symbol is included in the registered **calling** number in S28, before and after the symbol, counting is performed by another counter. For example, as shown in FIG. 8, when the registered **calling** number is composed of an area code (01234), a symbol (-), a local exchange number (5), a symbol (-) and. .

US PAT NO: 5,014,192 [IMAGE AVAILABLE] L25: 7 of 22
US-CL-CURRENT: 707/1; 340/825.05; 364/222.81, 232.1, 240.8,
241.1, 242.95, 280, 280.6, 281.3, 281.7, 281.8, 284,
284.4, 285, 286, DIG.1; 370/452

DETDESC:

DETD(15)

17-124 **Search** the file **directory** and find the
directory node associated with the supplied
<-----User Break----->

pathname. Also parse the file.sub.-- system name and
return it. Also, if the function is to create
a new file or **directory** then return that new
name in 'file'. If the **search** was unsuccessful
and the file system name was supplied, then
return an error to the user with the message
"File not found". If the **search** was
unsuccessful an

<-----User Break----->

'vol.sub.-- init', line 897-979.

153-236 Create a new data file.

161-164 Make sure the referenced **directory** in which the
new data file will be created is indeed a
directory (and not itself a data file).

165-171 If the name of the file to be created was
supplied, then see if it already exists in this
directory. If it does and is open then return
th

<-----User Break----->

name).

200-207 Build the new **directory** node.

208 Write the new **directory** node.

209-221 The write was successful. Build the reply
message to the user and send it.

223-234 The write. . . disk file and either error reply to
the user or forward the request.

242-287 Rename an existing file.

250 **Search** for the file to be renamed. Note.
Function 'findir' above has already found the
directory in which this file should be
found.

251-255 The file was found. If it is an open data file
then error reply the user with "File is
currently open".

256 **Search** for the **directory** in which the new name
is to be installed.

258-261 If not found error reply with "Rename **directory**
not found".

262-265 If it is not a **directory** error reply with "Not
a **directory**".

266-269 If file rename not supplied then error reply
with "Must specify rename file name".

270 **Search** for the file we are renaming to.

271-274 If it exists then error reply with "Rename file
already exists".

275-278 Delete the old **directory** node and create and
write the new **directory** node.

280-285 Rename **directory** could not be found. Either
error reply or forward the request.

292-311 Delete a data file.

300-303 If the specified file is a **directory** error reply
with "File is a **directory**".

304-307 If specified file is open error reply with

"File is currently open".
308 Delete the file.
317-361 Open a data file.
325-328 If specified file is a **directory** error reply
with "File is a **directory**".
329-334 If supplied access mode is not one of
"exclusive", "readwrite" or "readonly" then
error reply with "invalid. . .
"readwrite" then error reply with "File is
open; modes incompatible".
341-342 Set the requested access mode in the **directory**
entry.
343 **Increment** the access count.
344 If the access mode is not "readonly" then save
the pid of the user.
345 Rewrite the **directory** entry.

US PAT NO: 4,453,217 [IMAGE AVAILABLE] L25: 9 of 22
US-CL-CURRENT: 707/5; 364/222.2, 222.3, 222.81, 225, 225.4, 243,
243.7, 244, 244.3, 244.4, 251.6, 252, 252.1, 253, 253.1,
254, 254.3, 256.8, 259, 259.2, 259.6, 261.3, 261.5,
261.9, 262, 262.3, 263.3, 265, 266.6, 282.1, 282.3, 285,
285.3, 400, DIG.1

ABSTRACT:

There is disclosed a spelling correction arrangement for use in **directory** lookup applications. The arrangement corrects errors by finding the name in the **directory** that most closely resembles the name requested by the user. The arrangement is based on a recursive routine that continually subdivides the problem of finding a given name in a given **directory** into smaller subproblems in which shorter names are to be found in smaller directories. Multiple spelling errors are easily accommodated since the technique uses the given **directory** of
<-----User Break----->

names to limit the **search**. The technique allows the algorithm to find the closest name in the **directory** without actually considering the vast majority of the names that appear in the **directory**.

DETDESC:

<-----User Break----->

DETD(3)

The **directory** assistance algorithm uses a binary **search** based strategy to find the name (or character string) in the **directory** that most closely matches the name requested by the user. Strings are "close" if one can be obtained from the. . . be transformed into the other by a small number of character insertions or deletions. The algorithm uses an alphabetically ordered **directory** and a recursive function **called** FIND to **search** that **directory**. FIND **calls** itself recursively in a number of places to build and **search** a problem solving tree. The algorithm does not actually build an explicit tree but by virtue of the recursion "builds". . .

DETDDESC:

US PAT NO: 4,255,796 [IMAGE AVAILABLE] L25: 10 of 22
US-CL-CURRENT: 707/3; 364/926.1, 926.5, 927.2, 927.4, 928, 928.1,
929.2, 933.9, 934, 936.1, 937.1, 937.2
<-----User Break----->
2, 952.1, 956, 956.1, 958, 958.1,
958.2, 959.1, 963, 963.1, 964, 964.2, 964.5, 965, 965.5,
966.1, 966.4, 974, 974.1, 974.2, 974.5, 974.6, DIG.2

ABSTRACT:

An associative information retrieval system accepts information from a user and generates a query mask utilizing nested superimposed code words to **search** through and to find partial matches with the content of an auxiliary store. The auxiliary store contains similarly generated code. . . and the user is fed back information on the number of possible matches. The feedback informs the user on the **incremental** progress of the **search** produced in response to each newly entered character and also as part of a sequence that it may form with previously entered characters. The feedback information helps the user direct the **search** which the person does by supplying additional characters. When the number of possible matches is reduced to a manageable list, . . .

DETDESC:

DETD(24)

In . . . inventive principles. At keyboard 211 of terminal 210, the user inputs to the system a sequence of alphanumeric characters also **called** match specifications. This input is applied to SCW generator

US PAT NO: 4,104,717 [IMAGE AVAILABLE] L25: 12 of 22
US-CL-CURRENT: 707/5; 364/222.2, 222.3, 224, 224.1, 225, 225.4,
227.4, 234, 234.1, 235, 235.6, 236.3, 236.5, 238, 238.3,
238.4, 239, 239.4, 245.5, 245.9, 246, 246.2, 248, 248.2,
248.3, 252, 252.3, 252.4, 253, 253.1, 256.3, 256.4,
256.5, 259, 259.2, 265, 266.5, DIG.1

ABSTRACT:

In . . . primary name, secondary name, primary address, secondary address, generic occupation and specific occupation. The entire contents of a local telephone **directory** may be electronically stored at memory addresses based upon the above encoding. Statistical methods are used to ensure a workable. . . convertible into address coordinates of a random access core storage to speed information retrieval. Electronic calculating and comparing circuits automatically **search** the memory as a function of the division points, and also provide rapid and automatic
<-----User Break----->

address range computations. The system. . .

ETD(88)

As . . . outputs from gate 209' restore the y register 220 and x register 222 to their initial value of 1, clear **counter** 209 to its initial value logic 0, and clear comparator 272. **Counter** 209 keeps track of how many steps of comparison (how many stages of x of division <-----User Break----->

point analysis) have been. . . steps, when $x = m$, the last possible stage of division point analysis has been reached, because the division point **search** group only encompasses the first $2.\text{sup}.m$ tracks of the file memory. In the practical example given, $m = 10$ and. . .

US PAT NO: 4,077,059 [IMAGE AVAILABLE] L25: 13 of 22
US-CL-CURRENT: 711/122; 364/228.1, 236.2, 236.3, 236.4, 236.6, 238.4,
243, 243.1, 243.4, 245, 245.2, 246, 246.1, 246.2, 246.3,
248.1, 248.2, 251, 251.1, 251.3, 255.1, 255.7, 256.3,
256.4, 256.5, 259, 259.4, 260, 260.2, 265, 265.3, 268,
268.3, 268.5, 273.4, DIG.1; 707/202

ABSTRACT:

<-----User Break----->

This . . . The memory system has two different types of memory units on each level. One of the types of units is **called** the data store (DS) and contains all the data at that level of the memory. The other type of unit is **called** the copy back data store (CBDS) and contains

As . . . 1 supplies a write control signal to gate 66 to gate the output of the hash circuit 52 into the **search** ad

<-----User Break----->

search comparing the virtual address with that of each of the virtual addresses stored in the journal. If the virtual address in the **search** address register matches one in the journal, the journal 20a provides a compare or hit signal indicating that the address. . . copyback store determines the order in which that change is copied back into the lower levels. For this purpose a **counter** referred to as the next free space **counter** 70 counts off each of the addresses of the journal 20a and CBDS 18a in sequence. When a no-hit signal. . . is provided by the hit register of journal 20a in combination with the PPU write signal, the output of the **counter** 70 is fed through AND gates 72 to select any of the m words in the journal 20 and CBDS 18. Simultaneously the virtual address in the **search** register 68 is entered through AND gate 74 into the journal and the address and data from the PPU 10a are entered into the CBDS 18a at the location specified by the **counter** 70. Of course the change in data is also entered into the L1 store 14a.

D

d his

(FILE 'USPAT' ENTERED AT 19:19:51 ON 17 FEB 1998)

L1 140 S CALL? AND PERSONAL DIGITAL ASSISTANT
L2 18 S L1 AND DIRECTORY
L3 10 S L2 AND SEARCH
L4 754 S SEARCH (P) DIRECTORY
L5 3 S L4 AND L3
L6 754 S DIRECTORY (P) SEARCH
L7 2153 S SEARCH (P) INCREMENT?
L8 69 S L6 AND L7
L9 3634 S SEARCH (P) CHECK?
L10 654 S L9 AND L7
L11 34 S L9 AND L8
L12 34 S L10 AND L6
L13 1 S L10 AND PERSONAL DIGITAL ASSISTANT
L14 1 S L8 AND PERSONAL DIGITAL ASSISTANT
L15 57 S L8 AND CALL?
L16 12424 S SEARCH AND COUNTER
L17 13261 S L16 OR L7
L18 2518 S SEARCH (P) COUNTER
L19 843 S L18 AND L7
L20 3828 S L18 OR L7
L21 2349 S L20 AND CALL?
L22 80 S L6 AND L21
L23 0 S L22 AND CELLULAR
L24 41 S L22 AND (707/CLAS OR 455/CLAS OR 340/CLAS OR 370/CLAS OR
71
L25 22 S L22 AND (707/CLAS OR 455/CLAS OR 340/CLAS OR 370/CLAS OR
37

=> d 1-

1. 5,675,324, Oct. 7, 1997, Paging device having last-to-first sequential character memory search routine; Shintaro Hashimoto, et al., ~~340/825.44~~; 235/383; 340/311.1; 455/31.2 [IMAGE AVAILABLE]
2. 5,659,742, Aug. 19, 1997, Method for storing multi-media information in an information retrieval system; James T. Beattie, et al., 707/104 [IMAGE AVAILABLE]
3. 5,488,409, Jan. 30, 1996, Apparatus and method for tracking the playing of VCR programs; Henry C. Yuen, et al., 348/5, 2; 360/27, 72.2; 386/83; 455/2 [IMAGE AVAILABLE]
4. 5,437,029, Jul. 25, 1995, Path name resolution method providing fixed speed of file accessing in computer network; Pradeep K. Sinha, 707/200; 364/282.4, 284.4, 974.7, DIG.1, DIG.2; 395/200.75; 711/200 [IMAGE AVAILABLE]
5. 5,379,422, Jan. 3, 1995, Simple random sampling on pseudo-ranked hierarchical data structures in a data processing system; Gennady Antoshenkov, 707/1; 364/252, 282.1, DIG.1 [IMAGE AVAILABLE]
6. 5,053,945, Oct. 1, 1991, System and method for performing a multi-file transfer operation; John W. Whisler, 707/200; 364/239.6, 243.2, 245.6, 245.7, 246, 251.6, 254.3, 256.3, 256.8 [IMAGE AVAILABLE]

7. 5,014,192, May 1991, System for locating a file in a logical ring by sequentially forwarding access request with file system name and file name; Bruce M. Mansfield, et al., **707/1; 340/825.05; 364/222.81, 232.1, 240.8, 241.1, 242.95, 280, 280.6, 281.3, 281.7, 281.8, 284, 284.4, 285, 286, DIG.1; 370/452** [IMAGE AVAILABLE]
8. 4,989,132, Jan. 29, 1991, Object-oriented, logic, and database programming tool with garbage collection; Fredric H. Mellender, et al., 395/705; 364/274, 274.1, 280, 280.4, 281.1, 282.1, DIG.1; 395/76, 704; **707/103** [IMAGE AVAILABLE]
9. 4,453,217, Jun. 5, 1984, Directory lookup method and apparatus; Richard H. Bolvie, **707/5; 364/222.2, 222.3, 222.81, 225, 225.4, 243, 243.7, 244, 244.3, 244.4, 251.6, 252, 252.1, 253, 253.1, 254, 254.3, 256.8, 259, 259.2, 259.6, 261.3, 261.5, 261.9, 262, 262.3, 263.3, 265, 266.6, 282.1, 282.3, 285, 285.3, 400, DIG.1** [IMAGE AVAILABLE]
10. 4,255,796, Mar. 10, 1981, Associative information retrieval continuously guided by search status feedback; John D. Gabbe, et al., **707/3; 364/926.1, 926.5, 927.2, 927.4, 928, 928.1, 929.2, 933.9, 934, 936.1, 937.1, 937.2, 937.4, 939, 939.5, 940, 942, 942.7, 946.2, 947, 947.1, 947.2, 950, 950.4, 951.1, 951.4, 952, 952.1, 956, 956.1, 958, 958.1, 958.2, 959.1, 963, 963.1, 964, 964.2, 964.5, 965, 965.5, 966.1, 966.4, 974, 974.1, 974.2, 974.5, 974.6, DIG.2** [IMAGE AVAILABLE]
11. 4,117,542, Sep. 26, 1978, Electronic pocket directory; Judah Klausner, et al., 364/705.05; 341/23; 345/169, 172; 364/705.06, 919.4, 920, 926, 927, 927.2, 927.5, 927.8, 928, 928.1, 928.2, 929, 929.1, 931, 934, 934.1, 935, 935.2, 935.4, 935.6, 937.1, 944.4, 948.1, 949.3, 951.1, 951.4, 953, 953.1, 954, 954.1, 954.2, 954.3, 962, 962.1, 963, 963.1, 964, 965, 965.5, 965.8, DIG.2; **379/355, 444** [IMAGE AVAILABLE]
12. 4,104,717, Aug. 1, 1978, Automatic system for providing telephone number information service; Eiji Fujimura, **707/5; 364/222.2, 222.3, 224, 224.1, 225, 225.4, 227.4, 234, 234.1, 235, 235.6, 236.3, 236.5, 238, 238.3, 238.4, 239, 239.4, 245.5, 245.9, 246, 246.2, 248, 248.2, 248.3, 252, 252.3, 252.4, 253, 253.1, 256.3, 256.4, 256.5, 259, 259.2, 265, 266.5, DIG.1** [IMAGE AVAILABLE]
13. 4,077,059, Feb. 28, 1978, Multi-processing system with a hierarchical memory having journaling and copyback; Vincent A. Cordi, et al., 711/122; 364/228.1, 236.2, 236.3, 236.4, 236.6, 238.4, 243, 243.1, 243.4, 245, 245.2, 246, 246.1, 246.2, 246.3, 248.1, 248.2, 251, 251.1, 251.3, 255.1, 255.7, 256.3, 256.4, 256.5, 259, 259.4, 260, 260.2, 265, 265.3, 268, 268.3, 268.5, 273.4, DIG.1; **707/202** [IMAGE AVAILABLE]
14. 4,020,473, Apr. 26, 1977, Automatic system for providing telephone number information service; Eiji Fujimura, **707/3; 364/222.81, 222.82, 228.3, 234, 235, 237, 243, 243.1, 248.2, 252.3, 252.4, 259, 259.2, 268, 268.3, 268.4, DIG.1** [IMAGE AVAILABLE]
15. 4,020,466, Apr. 26, 1977, Memory hierarchy system with journaling and copy back; Vincent Anthony Cordi, et al., **707/201; 364/222.81, 222.82, 239, 239.4, 251, 251.3, 251.6, 253, 253.1, 254, 254.2, 254.6, 256.3, 256.4, 259, 259.2, 259.4, 260, 260.2, 264, 264.6, 268, 268.1, 268.3, 268.5, DIG.1; 711/143** [IMAGE AVAILABLE]
16. 3,928,724, Dec. 23, 1975, Voice-actuated telephone directory-assistance system; Harold E. Byram, et al., 704/275; **379/80, 89** [IMAGE AVAILABLE]
17. 3,916,387, Oct. 28, 1975, Directory searching method and means; Luther J. Woodrum, **707/3; 364/244, 244.6, 246, 246.3, 246.6, 246.9, 251, 251.6, 252, 252.2, 252.3, 252.4, 260.4, 260.6, 262, 262.1, 262.4,**

262.8, 270, 270.1, 270.1 [IMAGE AVAILABLE]

18. 3,916,112, Oct. 28, 1975, Stored program control with memory work area assignment in a communication switching system; Charles A. Kalat, et al., 379/244, 284 [IMAGE AVAILABLE]

19. 3,835,260, Sep. 10, 1974, COMMUNICATION SWITCHING SYSTEM, WITH MARKER, REGISTER, AND OTHER SUBSYSTEMS COORDINATED BY A STORED PROGRAM CENTRAL PROCESSOR; Kenneth E. Prescher, et al., 379/237, 269, 273, 279, 290, 302 [IMAGE AVAILABLE]

20. 3,825,689, Jul. 23, 1974, MESSAGE METERING AND STORAGE SYSTEM; James R. Baichtal, et al., 379/13, 125, 127 [IMAGE AVAILABLE]

21. 3,806,804, Apr. 23, 1974, RADIO TELEPHONE SYSTEM HAVING AUTOMATIC CHANNEL SELECTION; Lawrence W. Mills, et al., 455/509; 340/825.72; 455/517 [IMAGE AVAILABLE]

22. 3,749,844, Jul. 31, 1973, STORED PROGRAM SMALL EXCHANGE WITH REGISTERS AND SENDERS; John Peter Dufton, 379/112, 284, 288 [IMAGE AVAILABLE]

d his

(FILE 'USPAT' ENTERED AT 19:19:51 ON 17 FEB 1998)

L1 140 S CALL? AND PERSONAL DIGITAL ASSISTANT
L2 18 S L1 AND DIRECTORY

=> d 1

1. 5,715,395, Feb. 3, 1998, Method and apparatus for reducing network resource location traffic in a network; Roy Frank Brabson, et al., 1/1 [IMAGE AVAILABLE]
2. 5,689,547, Nov. 18, 1997, Network **directory** methods and systems for a cellular radiotelephone; Anders Lennart Molne, 379/379, 201 [IMAGE AVAILABLE]
3. 5,678,179, Oct. 14, 1997, Message transmission system and method for a radiocommunication system; Joseph Eric Turcotte, et al., 340/825.44; 395/188.01 [IMAGE AVAILABLE]
4. 5,657,258, Aug. 12, 1997, Mobile pen computer having an integrated palm rest; Anthony James Grewe, et al., 364/708.1; 345/173; 361/681 [IMAGE AVAILABLE]
5. 5,606,594, Feb. 25, 1997, Communication accessory and method of telecommunicating for a PDA; David S. Register, et al., 455/550; 379/357, 428, 434, 447; 455/90, 347 [IMAGE AVAILABLE]
6. 5,604,492, Feb. 18, 1997, Apparatus and method for **directory** linked canned pager messages; Ahmad H. Abdul-Halim, 340/825.44, 311.1; 379/142; 455/32.1 [IMAGE AVAILABLE]
7. 5,602,903, Feb. 11, 1997, Positioning system and method; Frederick W. LeBlanc, et al., 455/456; 342/450, 457 [IMAGE AVAILABLE]
8. 5,586,169, Dec. 17, 1996, Method of ringing a **call** responsive apparatus which answered an immediately preceding incoming **call**; Deborah L. Pinard, et al., 379/82, 201, 211, 233, 265 [IMAGE AVAILABLE]
9. 5,584,025, Dec. 10, 1996, Apparatus and method for interactive communication for tracking and viewing data; Ronald D. Keithley, et al., 707/104; 364/225.4, DIG.1 [IMAGE AVAILABLE]
10. 5,572,573, Nov. 5, 1996, Removable user interface for use with interactive electronic devices; Loren M. Sylvan, et al., 455/556; 341/23; 345/173; 364/709.1; 455/558 [IMAGE AVAILABLE]
11. 5,570,465, Oct. 29, 1996, Apparatus, method and system for printing of legal currency and negotiable instruments; Peter J. Tsakanikas, 395/114; 379/93.02; 395/117 [IMAGE AVAILABLE]
12. 5,570,412, Oct. 29, 1996, System and method for updating a location databank; Frederick W. LeBlanc, 455/456, 560 [IMAGE AVAILABLE]
13. 5,564,070, Oct. 8, 1996, Method and system for maintaining processing continuity to mobile computers in a wireless network; Roy Want, et al., 455/507; 370/341, 401 [IMAGE AVAILABLE]

14. 5,559,707, Sep. 24, 1996, Computer aided routing system; David M. DeLorme, et al., 701/200; 340/990, 995 [IMAGE AVAILABLE]

15. 5,544,222, Aug. 6, 1996, Cellular digital packet data mobile data base station; Mark T. Robinson, et al., 455/557; 370/422; 455/561 [IMAGE AVAILABLE]

16. 5,533,029, Jul. 2, 1996, Cellular digital packet data mobile data base station; Steven H. Gardner, 370/329, 412; 455/466 [IMAGE AVAILABLE]

17. 5,517,407, May 14, 1996, Device for including enhancing information with printed information and method for electronic searching thereof; Michael L. Weiner, 704/1; 707/1 [IMAGE AVAILABLE]

18. 5,493,105, Feb. 20, 1996, Electronic business card system; Nimesh R. Desai, 235/375 [IMAGE AVAILABLE]